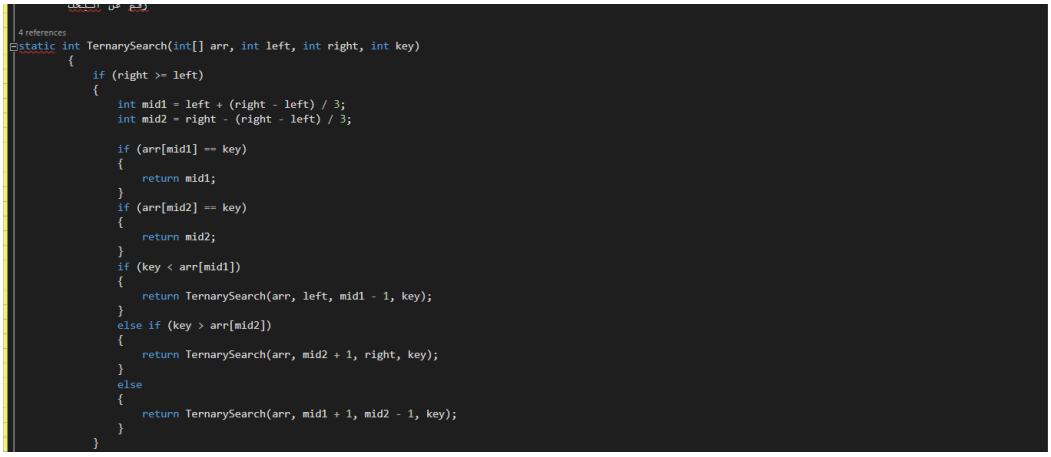
حل وظيفة البحث :



```
4 references
static int TernarySearch(int[] arr, int left, int right, int key)
        {
            if (right >= left)
            {
                int mid1 = left + (right - left) / 3;
                int mid2 = right - (right - left) / 3;

                if (arr[mid1] == key)
                {
                    return mid1;
                }
                if (arr[mid2] == key)
                {
                    return mid2;
                }
                if (key < arr[mid1])
                {
                    return TernarySearch(arr, left, mid1 - 1, key);
                }
                else if (key > arr[mid2])
                {
                    return TernarySearch(arr, mid2 + 1, right, key);
                }
                else
                {
                    return TernarySearch(arr, mid1 + 1, mid2 - 1, key);
                }
            }
```

```
            }
            return -1;
        }

        int[] arr = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
        int key = 5;
        int result = TernarySearch(arr, 0, arr.Length - 1, key);
if (result == -1)
{
    Console.WriteLine("Element not found");
}
else
{
    Console.WriteLine("Element found at index " + result);
}
```

```
static int CountOccurrence(int[] arr, int n)
{
    int left = 0;
    int right = arr.Length - 1;
    int first = -1;
    int last = -1;


    while (left <= right)
    {
        int mid = (left + right) / 2;

        if (arr[mid] == n)
        {
            first = mid;
            right = mid - 1;
        }
        else if (arr[mid] < n)
        {
            left = mid + 1;
        }
        else
        {
            right = mid - 1;
        }
    }


    left = 0;
    right = arr.Length - 1;
    while (left <= right)
    {
        int mid = (left + right) / 2;

        if (arr[mid] == n)
        {
            last = mid;
            left = mid + 1;
        }
```

```
        else if (arr[mid] < n)
        {
            left = mid + 1;
        }
        else
        {
            right = mid - 1;
        }
    }

    // Return count of n in arr[]
    if (first != -1 && last != -1)
    {
        return last - first + 1;
    }
    else
    {
        return 0;
    }
}

int[] arr = { 1, 2, 3, 4, 5, 5, 5, 5, 5, 6, 7, 8, 9, 10 };
Console.Write("enter number : ");
int key = int.Parse(Console.ReadLine());

int x = CountOccurrence(arr, key);
if (x == 0)
{
    Console.WriteLine("element not found ");
}
else
{

    Console.Write("number of times repeated :" + " " + x );
}
```